

Министерство науки и высшего образования РФ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Курский государственный университет»

Кафедра программного  
обеспечения и администрирования  
информационных систем

Направление подготовки  
математическое обеспечение и  
администрирование  
информационных систем

Форма обучения очная

**Отчет**

**по лабораторной работе №2.1**

«Программирование на языке C++ с использованием функций со сложными  
параметрами»

Выполнил:

студент группы 113.1

Козявин М. С.

Проверил:

старший преподаватель кафедры ПОиАИС

Ураева Е. Е.

Курск, 2021

**Цель работы:** Изучить особенности написания программ на языке C++ с использованием функций со сложными параметрами.

### **Задание**

**Задача 1.** Дан массив действительных чисел размера  $n$ . Выполнить сортировку элементов массива, расположенных между его максимальным по модулю и минимальным по модулю элементами, по убыванию, составив функцию, результатом которой является отсортированный массив. В качестве аргументов функции использовать параметры сортировки, указатель на массив и его размер.

**Задача 2.** Дана матрица  $A$  действительных чисел размера  $n \times m$ . Составив функцию, для поиска максимального элемента в указанной строке двумерного массива, сдвинуть в матрице  $A$  все строки циклически вправо на количество элементов равное максимальному элементу в этой строке.

**Задача 3.** Описать функцию  $\text{LinearSystemSolve}(A1, B1, C1, A2, B2, C2, X, Y)$ , решающую систему из двух линейных уравнений :  $A1 \cdot X + B1 \cdot Y = C1$ ,  $A2 \cdot X + B2 \cdot Y = C2$  и возвращающую результат через ссылки  $X$  и  $Y$  ( $A1, B1, C1, A2, B2, C2, X, Y$  – вещественные). Функция должна возвращать целое значение: 1, если существует ровно одно решение, 0, если система не имеет решений, и -1, если система имеет бесконечно много решений. С помощью этой функции найти решение двух систем линейных уравнений с двумя неизвестными, заданных двумя наборами из шести чисел :  $(A1, B1, C1, A2, B2, C2)$  и  $(A2, B2, C2, A3, B3, C3)$ . Вывести на экран значения корней или сообщения: «решений нет» или «решений бесконечно много».

### **Разработка алгоритма**

#### **Задача 1**

Входные данные:  $n$  - целое число,  $arr$  - массив из  $n$  действительных чисел

Выходные данные:  $arr$  - массив действительных чисел

Функция findMinMax находит максимальное и минимальное значение в строке n массива arr и записывает в переменные maxIndex и minIndex их индексы.

Входные аргументы:

arr – массив действительных чисел;

n, minIndex, maxIndex – целые числа

Функция sort сортирует массив arr между переданными индексами minIndex и maxIndex

Входные аргументы:

arr – массив действительных чисел;

minIndex, maxIndex – целые числа

Алгоритм решения задачи представлен на рисунке 1.

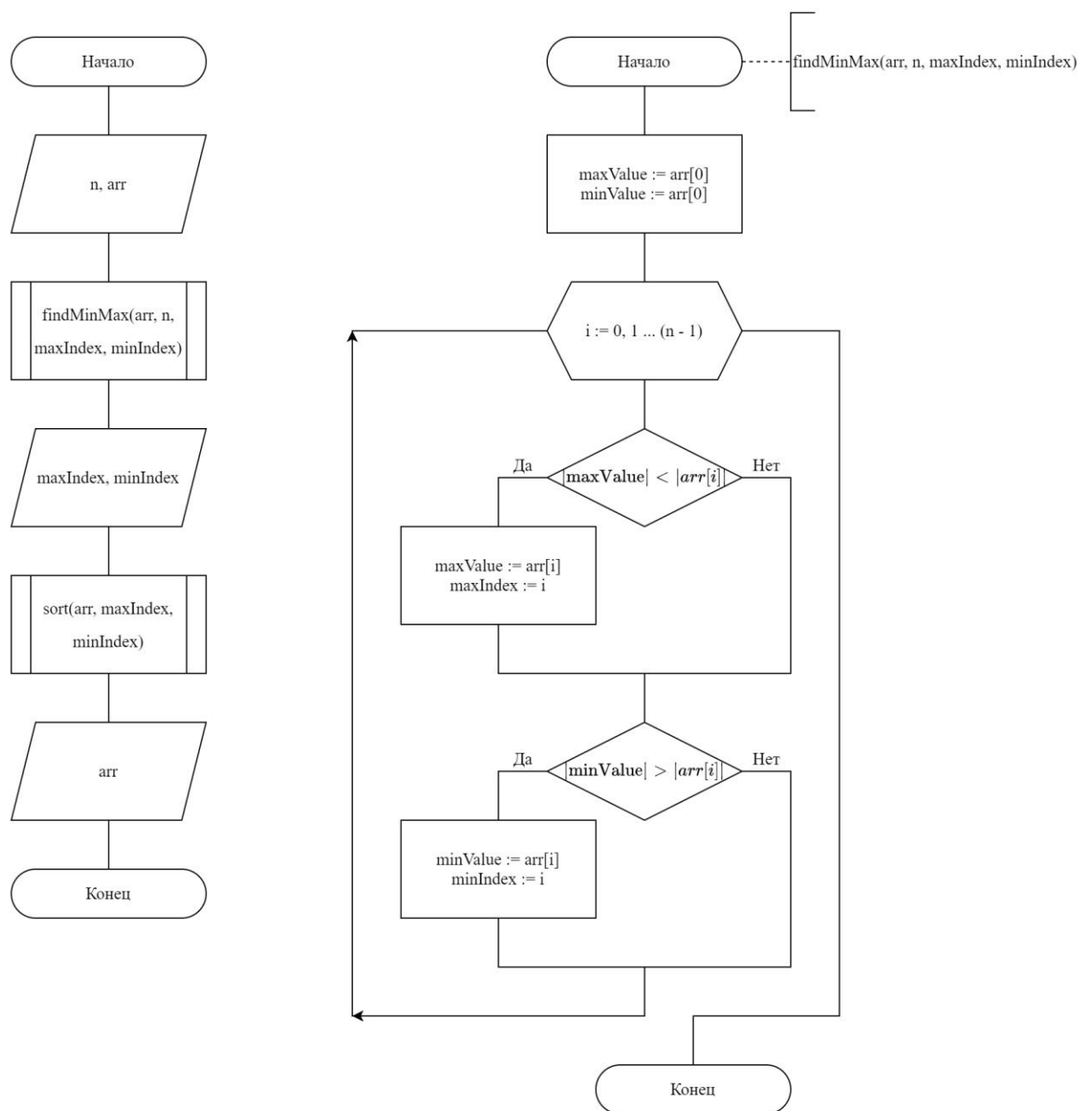


Рисунок 1 - Алгоритм решения задачи 1

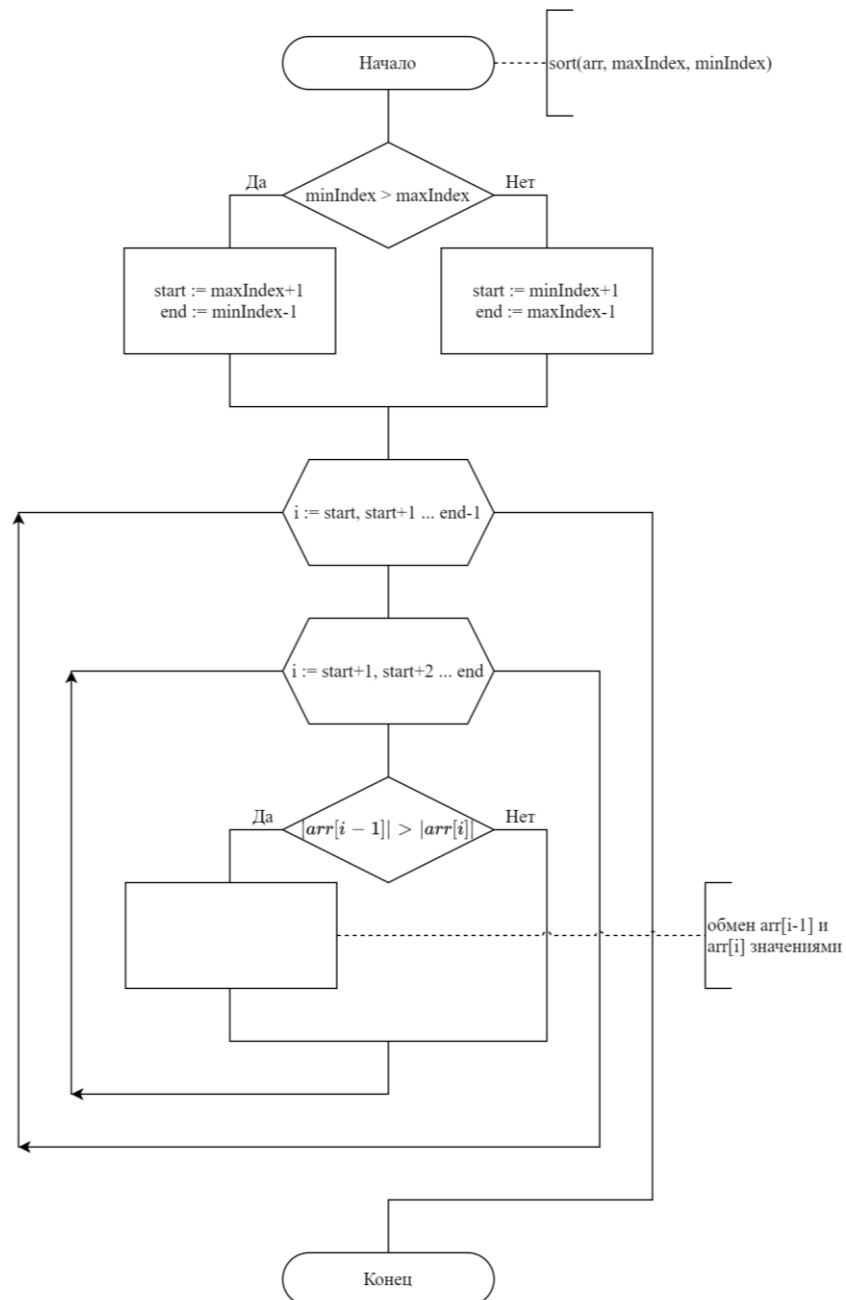


Рисунок 1 - Продолжение

### Задача 2

Входные данные:  $n$ ,  $m$  — целые числа,  $arr$  — массив из  $n*m$  действительных чисел.

Выходные данные:  $arr$  — массив действительных чисел.

Функция `shiftMatrix` производит сдвиг строчек матрицы

Входные данные:

*arr* – массив действительных чисел.

*n, m* – целые числа

Функция `findMaxInRow` находит максимальное значение в строчке *row* массива *arr*.

Входные данные:

*arr* – массив действительных чисел.

*row, m* – целые числа

Выходные данные:

*max* – действительное число

Функция `countElementsInRow` находит количество элементов *value* в строчке *row* массива *arr*.

Входные данные:

*arr* – массив действительных чисел.

*row, m* – целые числа

*value* – действительное число

Выходные данные:

*count* – целое число

Алгоритм решения задачи представлен на рисунке 2.

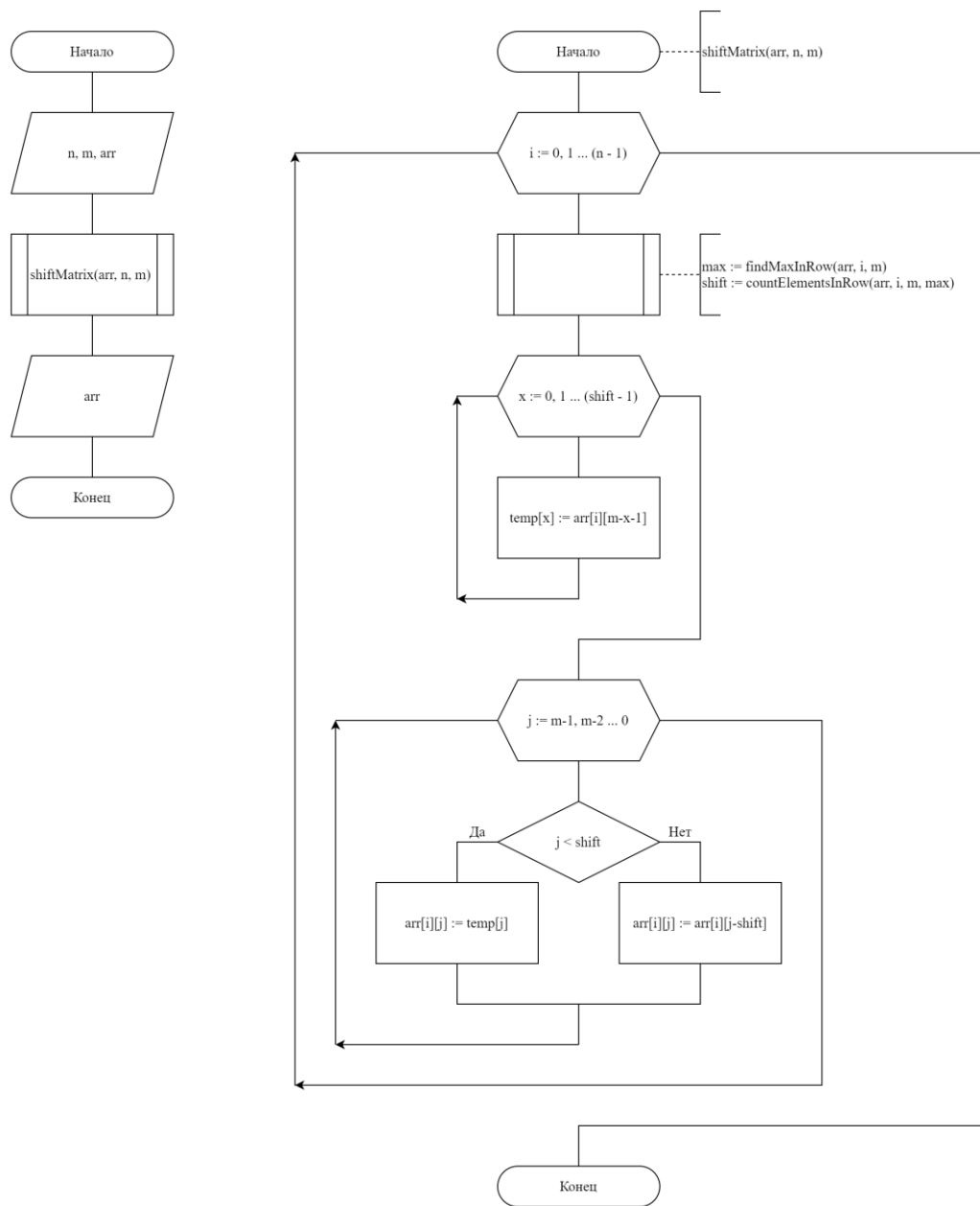


Рисунок 2 - Алгоритм решения задачи 2

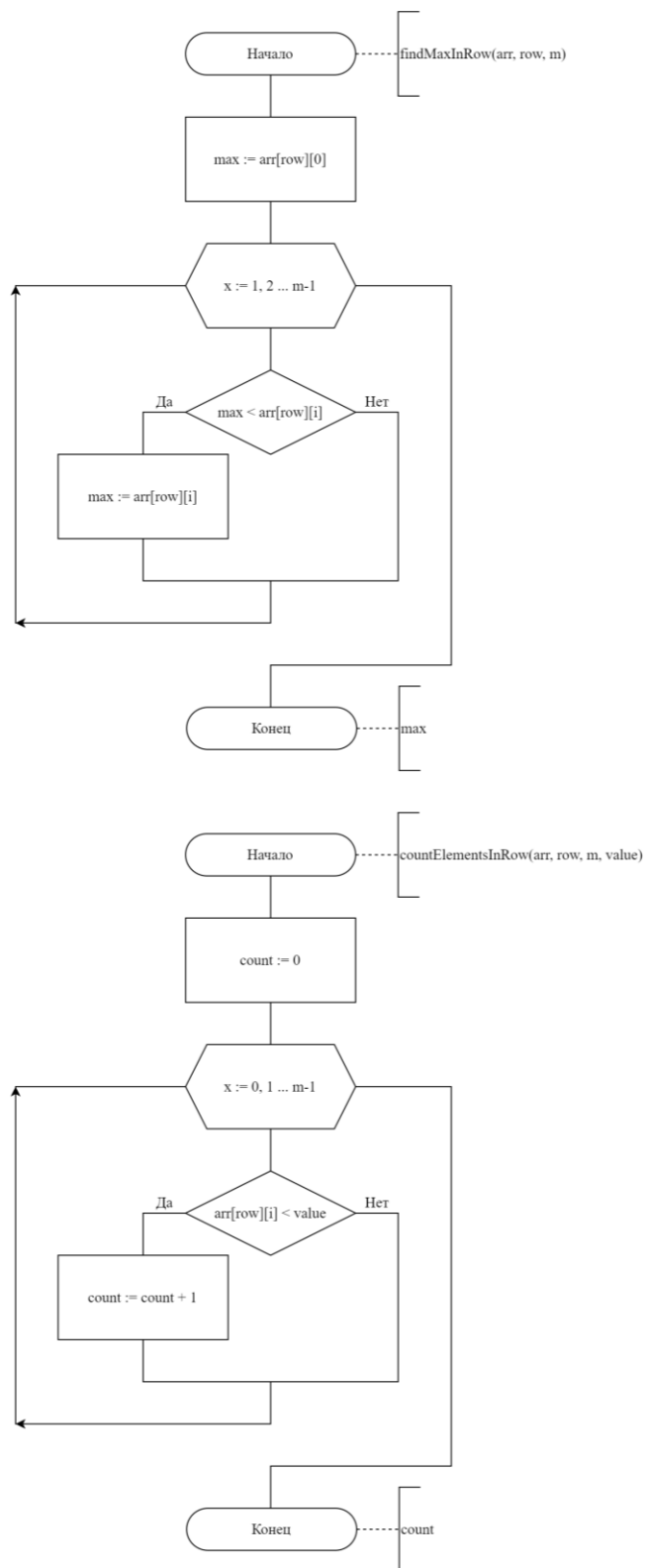


Рисунок 2 - Продолжение



### Задача 3

Входные данные:  $a1, b1, c1, a2, b2, c2$  – действительные числа

Выходные данные:  $x, y$  – действительные числа

Функция `solve` обрабатывает и выводит значение функции `LinearSystemSolve`

Входные данные:  $a1, b1, c1, a2, b2, c2, x, y$  – действительные числа

Функция `LinearSystemSolve` решает систему линейных уравнений и записывает результат в переменные  $x, y$

Входные данные:  $a1, b1, c1, a2, b2, c2, x, y$  – действительные числа

Алгоритм решения задачи представлен на рисунке 3.

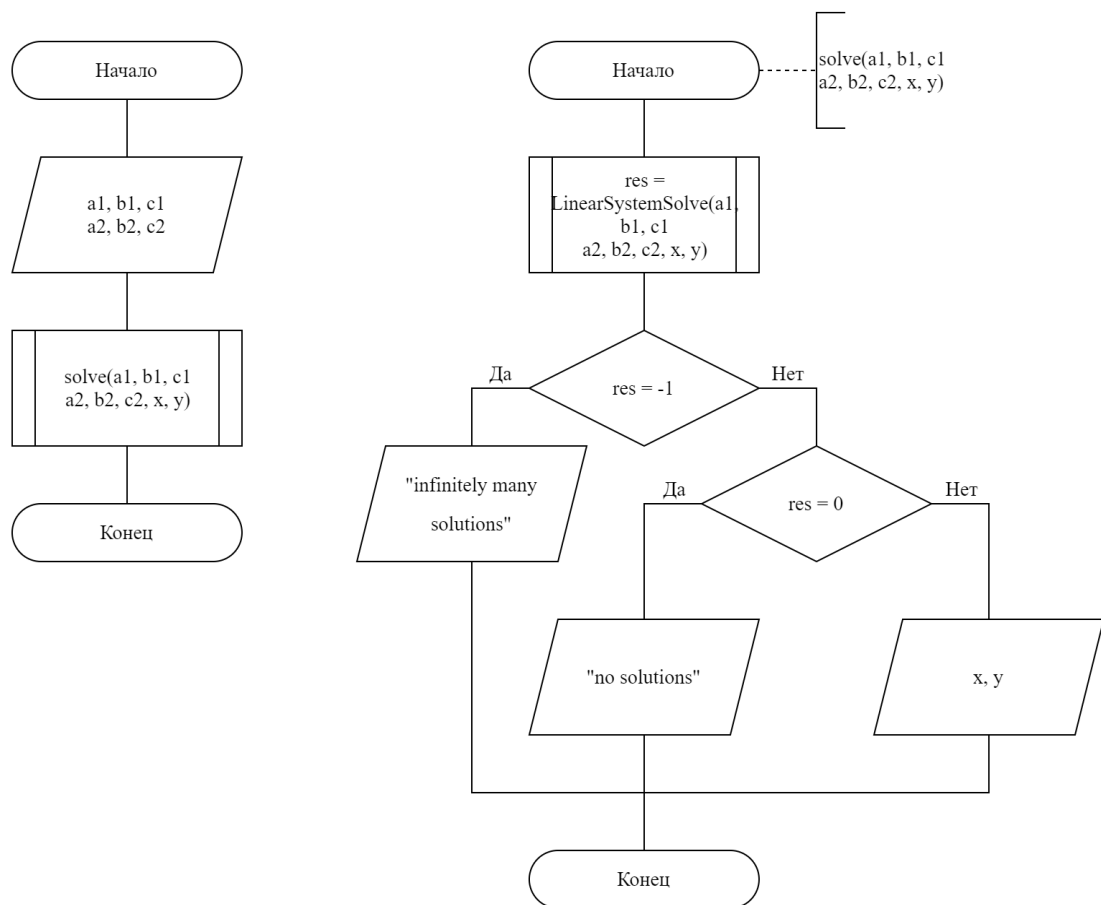


Рисунок 3 - Алгоритм решения задачи 3

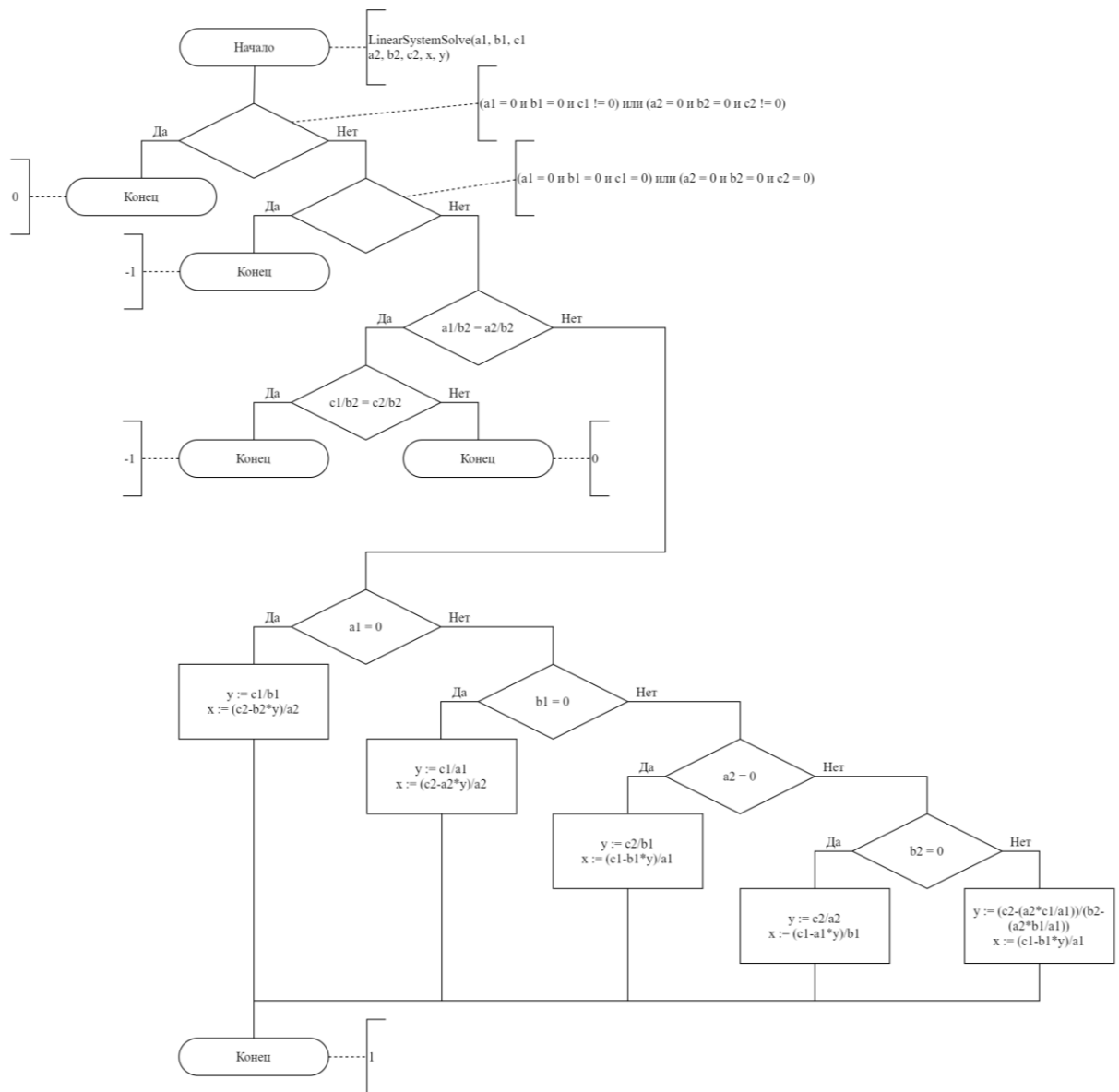


Рисунок 3 - Продолжение

### ***Текст программы***

#### *Текст программы для решения задач 1-3*

```

#include <iostream>
#include <cstdlib>
#include <ctime>
#include <cmath>
using namespace std;

```

```

void taskOne();

void findMinMax(double* arr, int n, int& maxIndex, int& minIndex);

void sort(double*& arr, int maxIndex, int minIndex);


void taskTwo();

double findMaxInRow(double **arr, int row, int m);

int countElementsInRow(double **arr, int row, int m, double value);

void shiftMatrix(double **arr, int n, int m);


void taskThree();

int LinearSystemSolve(double a1, double b1, double c1, double a2, double b2,
double c2, double &x, double &y);

void solve(double a1, double b1, double c1, double a2, double b2, double c2,
double &x, double &y);


int main() {
    int taskNumber;
    srand(time(NULL));

    cin >> taskNumber;

    while (!cin.eof()) {
        switch (taskNumber) {
            case 1:
                taskOne();
                cin >> taskNumber;
                break;

            case 2:
                taskTwo();

```

```

        cin >> taskNumber;
        break;

    case 3:
        taskThree();
        cin >> taskNumber;
        break;

    default:
        cout << "input error\n";
        cout << "ctrl + z for exit.\n";
        cin.clear();
        cin.ignore(1000, '\n');
        cin >> taskNumber;
        break;
    }
}

}

void taskOne() {
    int n, maxIndex, minIndex;
    cout << "input count of numbers: ";
    cin >> n;
    double *arr = new double[n];
    for (int i = 0; i < n; i++) {
        arr[i] = ( (rand() % 10000) / 100.0) - 50;
        cout << arr[i] << " ";
    }

    findMinMax(arr, n, maxIndex, minIndex);
}

```

```

cout << "\nMin index: " << minIndex << " max index: " << maxIndex << "\n";
sort(arr, maxIndex, minIndex);

for (int i = 0; i < n; i++) {
    cout << arr[i] << " ";
}
cout << "\n";
}

```

```

void findMinMax(double *arr, int n, int &maxIndex, int &minIndex) {
    double maxVal = arr[0], minVal = arr[0];
    for (int i = 0; i < n; i++) {
        if (abs(maxVal) < abs(arr[i])) {
            maxVal = arr[i];
            maxIndex = i;
        }

        if (abs(minVal) > abs(arr[i])) {
            minVal = arr[i];
            minIndex = i;
        }
    }
}

```

```

void sort(double*& arr, int maxIndex, int minIndex) {
    int start, end;

    if (minIndex > maxIndex) {
        start = maxIndex+1;
        end = minIndex-1;
    }
}

```

```

    } else {
        start = minIndex+1;
        end = maxIndex-1;
    }

    for (int x = start; x < end; x++) {
        for (int i = start+1; i <= end; i++) {
            if (abs(arr[i-1]) > abs(arr[i])) swap(arr[i-1], arr[i]);
        }
    }
}

void taskTwo() {
    int n = 3, m = 3;
    cout << "size of matrix: ";
    cin >> n >> m;

    double** arr = new double*[n];
    for (int i = 0; i < n; i++) {
        arr[i] = new double[m];
        for (int j = 0; j < m; j++) {
            cin >> arr[i][j];
        }
    }

    shiftMatrix(arr, n, m);

    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) {
            cout << arr[i][j] << " ";

```

```

    }
    cout << "\n";
}
}

void shiftMatrix(double **arr, int n, int m) {
    for (int i = 0; i < n; i++) {
        double max = findMaxInRow(arr, i, m);
        int shift = countElementsInRow(arr, i, m, max);

        double* temp = new double[shift];
        for (int x = 0; x < shift; x++) {
            temp[x] = arr[i][m - x - 1];
        }

        for (int j = m - 1; j >= 0; j--) {
            if (j < shift) {
                arr[i][j] = temp[j];
            } else {
                arr[i][j] = arr[i][j - shift];
            }
        }
    }
}
}

```

```

double findMaxInRow(double** arr, int row, int m) {
    double max = arr[row][0];
    for (int i = 1; i < m; i++) {
        if (max < arr[row][i]) {
            max = arr[row][i];
        }
    }
}

```

```

    }
}
return max;
}

```

```

int countElementsInRow(double** arr, int row, int m, double value) {
    int count = 0;
    for (int i = 0; i < m; i++) {
        if (arr[row][i] == value) {
            count++;
        }
    }
    return count;
}

```

```

void taskThree() {
    double a1, b1, c1, a2, b2, c2, x, y;
    //cout << "a1, b1, c1, a2, b2, c2: ";
    //cin >> a1 >> b1 >> c1 >> a2 >> b2 >> c2;
    solve(0, 0, 0, 0, 0, 0, x, y);
    solve(2, 5, 31, 1, 1, 8, x, y);
    solve(3, 10, 4, 6, 20, 8, x, y);
    solve(3, 10, 4, 6, 20, 13, x, y);

}

```

```

int LinearSystemSolve(double a1, double b1, double c1, double a2, double b2,
double c2, double &x, double &y) {

```

```

    if ((a1 == 0 && b1 == 0 && c1 != 0) || (a2 == 0 && b2 == 0 && c2 != 0)) {

```



```

    return 0;
}

if ((a1 == 0 && b1 == 0 && c1 == 0) || (a2 == 0 && b2 == 0 && c2 == 0)) {
    return -1;
}

if (a1 / b1 == a2 / b2) {
    if (c1/b1 == c2/b2) {
        return -1;
    } else {
        return 0;
    }
}

if (a1 == 0) {
    y = c1/b1;
    x = (c2-b2*y)/a2;
} else if (b1 == 0) {
    x = c1/a1;
    y = (c2-a2*x)/b2;
} else if (a2 == 0) {
    y = c2 / b2;
    x = (c1 - b1 * y) / a1;
} else if (b2 == 0) {
    x = c2 / a2;
    y = (c1 - a1 * x) / b1;
} else {
    y = (c2 - (a2 * c1 / a1)) / (b2 - (a2 * b1 / a1));

```

```

        x = (c1 - b1 * y) / a1;
    }

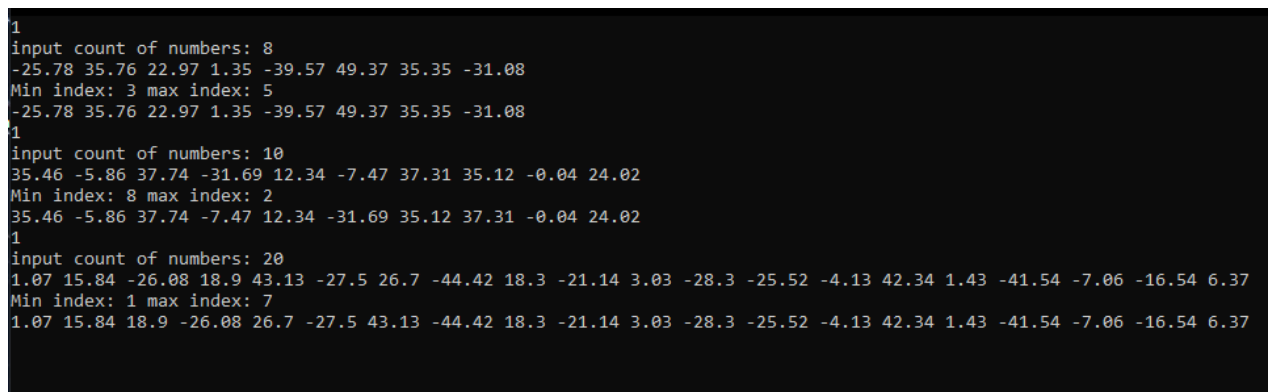
    return 1;
}

void solve(double a1, double b1, double c1, double a2, double b2, double c2,
double &x, double &y) {
    int res = LinearSystemSolve(a1, b1, c1, a2, b2, c2, x, y);
    if (res == -1) {
        cout << "infinitely many solutions\n";
    } else if (res == 0) {
        cout << "no solutions\n";
    } else {
        cout << x << " " << y << '\n';
    }
}
}

```

### ***Тестирование программы***

Тестирование задачи 1 представлено на рисунке 4



```

1
input count of numbers: 8
-25.78 35.76 22.97 1.35 -39.57 49.37 35.35 -31.08
Min index: 3 max index: 5
-25.78 35.76 22.97 1.35 -39.57 49.37 35.35 -31.08
1
input count of numbers: 10
35.46 -5.86 37.74 -31.69 12.34 -7.47 37.31 35.12 -0.04 24.02
Min index: 8 max index: 2
35.46 -5.86 37.74 -7.47 12.34 -31.69 35.12 37.31 -0.04 24.02
1
input count of numbers: 20
1.07 15.84 -26.08 18.9 43.13 -27.5 26.7 -44.42 18.3 -21.14 3.03 -28.3 -25.52 -4.13 42.34 1.43 -41.54 -7.06 -16.54 6.37
Min index: 1 max index: 7
1.07 15.84 18.9 -26.08 26.7 -27.5 43.13 -44.42 18.3 -21.14 3.03 -28.3 -25.52 -4.13 42.34 1.43 -41.54 -7.06 -16.54 6.37

```

Рисунок 4 - Тест 1-3 задачи 1

Тестирование задачи 2 представлено на рисунке 5

```
2
size of matrix: 4 4
1 2 2 3
4 1 1 1
1 2 3 4
9 8 9 1
3 1 2 2
1 4 1 1
4 1 2 3
1 9 9 8
2
size of matrix: 4 5
1 1 1 1 2
2 2 0 2 1
4 0 2 9 9
8 8 2 3 4
2 1 1 1 1
1 2 0 2 2
9 9 4 0 2
4 3 8 8 2
```

Рисунок 5 - Тест 1-2 задачи 2

Тестирование задачи 3 представлено на рисунке 5

```
3
infinitely many solutions
3 5
infinitely many solutions
no solutions
```

Рисунок 6 - Тест 1-4 задачи 3